

What is claimed is:

1. A method of managing resources for a software application deployed in a client-server system, comprising:
- (a) receiving at the server system a request from a client computer to change the software application;
  - (b) receiving into the client computer, from the server system, a preference parameter associated with a group of resources needed to accommodate the request;
  - (c) determining whether the needed group of resources is stored locally on the client computer; and
  - (d) retrieving the needed group of resources from the server system if the needed group of resources is not stored locally on the client computer.
2. The method of claim 1, further comprising:
- (e) passing the needed group of resources to the software application so as to enable the application to access the group of resources.
3. The method of claim 2, wherein step (e) comprises:
- calling a function exposed by the software application; and
  - passing the needed group of resources to the software application as an argument.
4. The method of claim 2, further comprising:
- (f) generating a user interface on the client computer.
5. The method of claim 4, wherein step (f) comprises:
- linking to the group of resources;
  - loading text strings and image data into memory of the client computer; and

Sub  
ai

producing a user interface using the text strings and image data.

6. The method of claim 1, further comprising:

- (e) storing the needed group of resources in a memory space of the client computer; and
- (f) passing the needed group of resources to the software application so as to enable the application to access the group of resources.

7. The method of claim 6, further comprising:

- (g) generating a user interface on the client computer.

Sub  
ai

8. A method of internationalizing a software application deployed in a client server environment, comprising:

- (a) storing on a server system a plurality of geographic region-specific groups of resources;
- (b) receiving at a server system a request from a client computer that the software application exhibit characteristics indicative of a desired geographic region;
- (c) receiving from the server system into a resource manager on the client computer a preference parameter associated with a region-specific group of resources needed to accommodate the request;
- (d) determining whether the needed region-specific group of resources is stored locally on the client computer; and
- (e) retrieving the region-specific group of resources from the server system if the needed region-specific group of resources is not stored locally on the client computer.

9. The method of claim 8, further comprising:

002090.5.029560

*Sub C*

(f) passing the needed region-specific group of resources to the software application so as to enable the application to access the needed region-specific group of resources.

10. The method of claim 9, wherein step (f) comprises:  
calling a function exposed by the software application; and  
passing the needed region-specific group of resources to the software application as an argument.

11. The method of claim 10, further comprising:

(g) generating a user interface on the client computer.

*Sub C*

12. The method of claim 11, wherein step (g) comprises:

linking to the group of resources;

loading text strings and image data into memory of the client computer; and

producing a user interface using the text strings and image data.

13. The method of claim 8, further comprising:

(f) storing the needed region-specific group of resources in a memory space of the client computer; and

(g) passing the needed region-specific group of resources to the software application so as to enable the application to access the needed region-specific group of resources.

14. The method of claim 13, further comprising:

(h) generating a user interface on the client computer utilizing the resources of the needed group.

Sub 02  
15. A system for customizing a user interface of a software application stored on a client computer; comprising:

a communication link to a server system; and

a processor of the client computer operative with a software implemented resource manager to:

receive from said server system a preference parameter associated with a needed group of resources;

determine whether the needed group of resources associated with the preference parameter is stored on the client computer; and

retrieve the needed group of resources associated with the preference parameter from the server system if the needed group of resources is not stored locally on the client computer so that the software application is enabled to generate a user interface of the software application using the needed group of resources.

16. The system of claim 15, wherein the client computer receives the preference parameter from the server system in response to a user-generated request to change a preference.

Sub 03  
17. The system of claim 16, wherein the preference is a regional setting.

18. The system of claim 16, wherein the preference is a language setting.

19. The system of claim 15, wherein the generation of the user interface further comprises calling a function exposed by the software application, passing the needed group of resources to the software application as an argument, so as to enable the software application to link to the needed group of resources, load text strings and image data into a memory of the client computer, and generate the customized user interface using the text strings and image data.

July  
10/3

20. A client-server system for permitting the internationalization of one or more software applications, comprising:

a server system having stored thereon a plurality of groups of resources, each of the plurality of groups associated with a preference, and the server system maintaining a user profile database for storing the preference of a user; and

a client computer interconnected with the server system through a network, the client computer having an input device for permitting the user to transmit preference indicative data to the server system and a software implemented resource manager for receiving from the server system a preference parameter associated with the group of resources corresponding to the preference parameter, such that the resource manager can determine whether the group of resources are stored locally on the client computer.

July  
10/3

21. The client-server system of claim 20, wherein the group of resources is retrieved from the server system if the group of resources is not stored locally on the client computer.

22. A method of modifying a toolbar interface of a browser application, comprising:

- (a) generating on a client computer the toolbar interface using a first group of resources;
- (b) receiving into a server system a request to change the toolbar interface;
- (c) communicating, from the server system, a unique identifier associated with a second group of resources needed to change the toolbar interface to the client computer;
- (d) determining whether the second group of resources associated with the unique identifier is stored locally on the client computer; and

Sub  
Sci

(e) retrieving the second group of resources associated with the unique identifier from the server system if the second group of resources is not stored locally on the client computer.

23. The method of claim 22, further comprising:

(f) passing the second group of resources to the browser application so as to enable the browser application to access the second group of resources.

24. The method of claim 23, wherein step (f) comprises:

calling a function exposed by the browser application; and

passing the group of resources to the browser application as an argument.

25. The method of claim 23, further comprising:

(g) accessing the second group of resources;

(h) loading text strings and image data of the second group of resources into a memory of the client computer; and

(i) generating the toolbar interface using the text strings and image data.

26. The method of claim 22, further comprising:

(f) storing the group of resources in a memory space of the client computer;

and

(g) passing the group of resources to the software application so as to enable the application to access the group of resources.

Sub  
Sci

27. ~~A method of increasing the desirability of a user accessible web site using a browser application and a client computer, the method comprising:~~

*Sub*  
*ad* (a) maintaining a user profile database on a server system for serving the web site to the browser application, the user profile database including at least one customizable option;

(b) permitting the user to change the customizable option; and

(c) generating a browser interface on the client computer in response to the change in the customizable option without the need to restart the browser application.

*Sub*  
*ad* 28. The method of claim 27, wherein step (c) comprises:  
communicating from the server system to the client computer a unique identifier associated with a group of resources needed to accommodate the change in the customizable option;

determining whether the group of resources associated with the unique identifier is stored locally on the client computer;

retrieving the group of resources associated with the unique identifier from the server system if the group of resources is not stored locally on the client computer; and

passing the group of resources to the browser application so as to enable the browser application to access the group of resources.

29. The method of claim 28 wherein step of passing the group of resources to the browser application comprises:

calling a function exposed by the browser application; and

passing the group of resources to the browser application as an argument.

30. A method of modifying a user's interaction with a software application deployed in a client server environment without the need to re-start the application, comprising the steps of:



- (a) identifying a user interface preference of the user;
- (b) identifying a preference specific resource necessary to the application to meet the user's preference; and
- (c) making the preference specific resource available to the application.
31. The method of claim 30, wherein step (a) further comprises:
- receiving into a server system a request to change a preference setting; and
- communicating to a client computer from the server system a preference parameter identifying the user interface preference.
32. The method of claim 31, wherein step (b) further comprises:
- retrieving a resource identifier associated with the preference specific resource identified by the preference parameter from a map of resource identifiers.
33. The method of claim 32, wherein step (c) further comprises:
- determining whether the preference specific resource is locally stored using the resource identifier;
- retrieving the preference specific resource from the server system, if the preference specific resource is not locally stored; and
- passing the preference specific resource to the software application as an argument.
34. The method of claim 30, wherein step (a) further comprises:
- receiving into a server system a request to change a preference setting; and
- receiving into a client computer a preference parameter identifying the user interface preference during a subsequent communication with the server system.



35. The method of claim 34, wherein step (b) further comprises:  
retrieving a resource identifier associated with the preference specific resource identified by the preference parameter from a map of resource identifiers.
36. The method of claim 35, wherein step (c) further comprises:  
determining whether the preference specific resource is locally stored using the resource identifier;  
retrieving the preference specific resource from the server system, if the preference specific resource is not locally stored; and  
passing the preference specific resource to the software application as an argument.
37. A method of adapting a user interface of a software application in a client server environment to a user's specific language requirements without the need to restart the application, comprising:  
(a) passing an item of data indicative of a user's language requirement to a server system;  
(b) identifying the resources necessary to generate the user interface in the user's required language; and  
(c) making the necessary resources available to the program.
38. The method of claim 37, wherein step (a) further comprises:  
inputting into a client computer a user identifier associated with the user's language requirement; and  
communicating the user identifier to the server system.

39. The method of claim 38, wherein step (b) further comprises:  
receiving into the client computer from the server system a preference parameter  
identifying the necessary resources; and  
retrieving a resource identifier associated with the necessary resources identified  
by the preference parameter from a map of resource identifiers.

40. The method of claim 39, wherein step (c) further comprises:  
determining whether the necessary resources are locally stored using the resource  
identifier;  
retrieving the necessary resources from the server system, if the necessary  
resources are not locally stored; and  
passing the necessary resources to the software application as an argument.

41. A method of dynamically generating a language-specific user interface of a  
software application deployed in a client server environment, comprising:  
(a) storing on a server system a plurality of language-specific resources;  
(b) receiving at a server system a request from a client computer that the  
software application generate the language-specific user interface;  
(c) receiving from the server system into the software application on the client  
computer a preference parameter associated with a language-specific resource needed to  
accommodate the request;  
(d) calling a function exposed by a resource manager instructing the resource  
manager to return the needed language-specific resource to the software application;  
(e) determining whether the needed language-specific resource is stored  
locally on the client computer;

(f) retrieving the needed language-specific resource from the server system if the needed language-specific resource is not stored locally on the client computer;

(g) passing the needed language-specific resource to the software application so as to enable the application to access the needed language-specific resource; and

(h) generating the language-specific user interface utilizing the language-specific resource.

42. The method of claim 41, wherein step (b) comprises:

accessing a user profile database;

changing a language setting stored in the user profile database; and

setting the preference parameter based upon the changed language setting.

43. The method of claim 41, wherein step (b) comprises:

logging onto the server system using a user identifier, the user identifier being associated with a language setting stored in a user profile database; and

setting the preference parameter based upon the stored language setting.

44. The method of claim 41, wherein step (g) comprises:

passing a pointer to the language-specific resource to the software application;

linking to the language-specific resource;

loading text strings and image data into memory of the client computer; and

producing the language-specific user interface using the text strings and image data.

45. A method of dynamically generating a language-specific user interface of a software application deployed in a client server environment, comprising:

Sub  
#1

- (a) storing on a server system a plurality of language-specific resources;
- (b) receiving at a server system a request from a client computer that the software application generate the language-specific user interface;
- (c) receiving from the server system into a resource manager on the client computer a preference parameter associated with a language-specific resource needed to accommodate the request;
- (d) determining whether the needed language-specific resource is stored locally on the client computer;
- (e) retrieving the needed language-specific resource from the server system if the needed language-specific resource is not stored locally on the client computer;
- (f) passing the needed language-specific resource to the software application so as to enable the application to access the needed language-specific resource; and
- (g) generating the language-specific user interface utilizing the language-specific resource.

46. The method of claim 45, wherein step (b) comprises:

- accessing a user profile database;
- changing a language setting stored in the user profile database; and
- setting the preference parameter based upon the changed language setting.

47. The method of claim 45, wherein step (b) comprises:

- logging onto the server system using a user identifier, the user identifier being associated with a language setting stored in a user profile database; and
- setting the preference parameter based upon the stored language setting.

Sub  
C1

48. The method of claim 45, wherein step (f) comprises:
- calling a function exposed by the software application; and
  - passing a pointer to the language-specific resource to the software application.
49. The method of claim 48, wherein step (g) comprises:
- linking to the language-specific resource using the pointer;
  - loading text strings and image data into memory of the client computer; and
  - producing the language-specific user interface using the text strings and image

data.

add  
a5

002090" 52028560